

Senioridade e continuando os estudos

Transcrição

Como você pode ver, existem algumas maneiras de trabalhar. Seja em uma maratona de programação ou outra competição, ou seja com um cliente, uma das coisas que diferencia muito um programador sênior de um que está começando é o momento de parar para pensar. É o refletir sobre aquilo que estamos fazendo.

Assim, se o cliente me pedir uma coisa, devemos olhar aquela coisa de diversos lados e tentar resolvê-la de diversas maneiras, pensando não apenas no caminho feliz, mas nos que podem trazer problemas para o código.

Talvez seja preciso conversar com o cliente, ou pedir para o examinador de uma prova esclarecer algum ponto do problema. Mas temos que olhar para ele de diversos ângulos.

Nós também programamos nos preocupando com a clareza do código, principalmente porque estamos trabalhando em equipe. Claro que, em uma maratona, poderíamos ser mais extremos com as variáveis. Porém, na minha experiência de maratonas, eu usava as variáveis da maneira que mostrei para vocês, com nomes razoavelmente claros, para que sempre saibamos o que está acontecendo no código naquele instante. Também é interessante extrair diversos métodos, principalmente trabalhando com Java. Com C talvez possamos extrair menos, de acordo com a situação. Tudo depende do caso.

O importante desse curso é que paremos para pensar no código que estamos criando e nos buracos que estamos deixando. Conversar para ver se é preciso cobrir os buracos, ou há alguns que podemos deixar. O que deveríamos fazer no caso do `else`, e não no do `if`? O que precisaríamos fazer nas bordas, ou seja, na condição inicial e na condição final. Tem `+1` ou `-1`? Tem igual ou não? Há chances de criarmos um loop infinito? É preciso parar para pensar sobre ele.

Todo laço pode entrar em loop infinito ou não rodar nunca. Exceto casos extremos de `while(true)` e `while(false)`, que podem ter problemas de compilação. Às vezes o próprio compilador aponta esse tipo de erro, mas lembre-se que todo laço exige um cuidado. Todo `if` precisa que pensemos no `else` e todo laço precisa que pensemos com calma e com carinho em suas condições inicial e final. Temos que analisar isso tudo pensando "Quando isso aqui vai quebrar?". Essa é a pergunta essencial.

Uma maratona é isso: considerar as possibilidades e identificar em quais delas o código não funcionará. E então, cuidar disso. E temos que fazer direito, pois se houver uma situação em que o código não funciona, o juiz diminuirá nossa nota na hora. E o competidor deixa de pegar medalha, ir para mundial, etc. Isso tudo por não pensar em um caso. Maratona é uma situação muito boa justamente por nos forçar a pensar em vários casos.

Evite problemas que exigem que você digite o mínimo possível, pois isso não existe no mundo real. São casos raríssimos em que esse tipo de exigência é pertinente. E não estamos falando em desenvolvimento mobile, em que "o programador precisa digitar menos porque o celular cabe menos". Também não é isso. Novamente: somos programadores, não digitadores. Estamos aqui para pegar um problema, pensá-lo de diversas maneiras e resolvê-lo de uma maneira estável para que ele não falhe. É esse o nosso trabalho, e é isso que eu espero que tenha aprendido nesse curso.

Para você que está em uma faculdade, e quer participar disso: procure a equipe de maratona de programação. Deve ter algum professor ou *coach* acessível. É possível encontrar também grupos de maratona de programação no Facebook e em outras redes.

Se você não estiver em uma faculdade, mas quer aprender mais e desenvolver mais seu pensamento nessa área, há diversos sites além dos dois que mostramos com uma infinidade de problemas de maratona. E os problemas começam em nível básico, para que você treine bastante. Eu pegava 100 problemas do mesmo tipo e os atacava, ora para treinar o input, ora para treinar a saída, ora para treinar os laços.

Não tem segredo: é treinando bastante que ficaremos bons. Não é treinar digitar pouco para construir um código, mas pensar no que pode dar errado em um código. Então vale a pena treinar mil vezes. Ao menos foi isso que fiz durante os meus 7 anos maratonas, e claro, nos meus 20 anos programando.

Na plataforma da Alura, temos diversos cursos que podem complementar o conhecimento obtido aqui. Eles lidam com questões relacionadas a algoritmos e pensar sobre problemas. Nossos cursos em geral já têm essa linha, porém há cursos específicos de [machine](https://cursos.alura.com.br/course/introducao-a-machine-learning-com-classificacao) (<https://cursos.alura.com.br/course/introducao-a-machine-learning-com-classificacao>), [learning](https://cursos.alura.com.br/course/machine-learning-introducao-aos-sistemas-de-recomendacoes) (<https://cursos.alura.com.br/course/machine-learning-introducao-aos-sistemas-de-recomendacoes>), [inteligência artificial](https://cursos.alura.com.br/career/inteligencia-artificial-iniciante) (<https://cursos.alura.com.br/career/inteligencia-artificial-iniciante>) e similares, em que falamos bastante de algoritmos. Há também cursos específicos de [algoritmos](https://cursos.alura.com.br/course/projetos-de-algoritmos-1) (<https://cursos.alura.com.br/course/projetos-de-algoritmos-1>), e cursos de estrutura de dados, que nos ajudam a abordar os problemas do nosso dia a dia e também de maratonas.

Tudo isso é fundamental para que ganhemos experiência como bons programadores, não como programadores que conhecem APIs. Como já vimos, conhecer APIs é importante para que não reinventemos a roda. Porém, saber pensar todos os casos de um problema, é essencial para não entregarmos um programa que não funciona. Não foque só em uma, pois as duas coisas são importantes. Desejo uma boa sorte nos seus próximos cursos e competições. Um abraço e até mais!